DESIGN OF A CHANNEL ERROR SIMULATOR USING VIRTUAL INSTRUMENT TECHNIQUES FOR THE INITIAL TESTING OF TCP/IP AND SCPS PROTOCOLS

Dr. Stephen Horan
Ru-hai Wang

April 1, 1999 **NMSU-ECE-99-002**

DESIGN OF A CHANNEL ERROR SIMULATOR USING VIRTUAL INSTRUMENT TECHNIQUES FOR THE INITIAL TESTING OF TCP/IP AND SCPS PROTOCOLS

Dr. Stephen Horan and Ru-hai Wang
Lujan Space Tele-Engineering Program
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Box 30001, MSC 3-O
Las Cruces, NM 88003-8001

New Mexico State University Technical Report
NMSU-ECE-99-002

April 1, 1999

Table of Contents

ACRONYM LIST

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CCSDS	Consultative Committee for Space Data Systems
Eb/No	Energy-per-bit-to-Noise-density Ratio
fp	File Protocol
ftp	File Transport Protocol
NMSU	New Mexico State University
PC	Personal Computer (Intel/Windows based configuration)
SCPS	Space Communications Protocol Specification
SGLS	Space-to-Ground Link Simulator
TCP/IP	Transmission Control Protocol/Internet Protocol
VI	Virtual Instrument
VME	Versa Module Eurocard

SECTION I - BACKGROUND

There exists a need for designers and developers to have a method to conveniently test a variety of communications parameters for an overall system design. This is no different when testing network protocols as when testing modulation formats. In this report, we discuss a means of providing a networking test device specifically designed to be used for space communications. This test device is a PC-based Virtual Instrument (VI) programmed using the LabVIEW™ version 5 [1] software suite developed by National Instruments™. This instrument was designed to be portable and usable by others without special, additional equipment. The programming was designed to replicate a VME-based hardware module developed earlier at New Mexico State University (NMSU) [2] and to provide expanded capabilities exceeding the baseline configuration existing in that module.

This report describes the design goals for the VI module in the next section and follows that with a description of the design of the VI instrument. This is followed with a description of the validation tests run on the VI. An application of the error-generating VI to networking protocols is then given.

SECTION II - DESIGN GOALS

The design of the Space-to-Ground Link Simulator (SGLS) for modeling satellite channel error scenarios was based on the following goals to replicate the statistical characteristics of a satellite channel:

a. Allow for simultaneous bi-directional data flow (forward and return channels);

b. Allow user-selectable error rates and statistical descriptions of the channel;

c. Allow time-variable error rates over several minutes as would be found in a satellite pass;

d. Allow different data rates on the forward and return links as would be found in satellite links, e.g. 2400 baud forward, 57600 baud;

e. Provide for a simulated ¼-second delay as typically found in satellite channels.

The first design goal is documented in this report. As additional modules are developed and tested, they will be individually documented to provide an overall VI architecture for the channel error simulator.

By using a PC-based configuration and not a generic networking simulator package, we believe the VI configuration to allow for several advantages, including:

a. Allowing tests on actual data streams with operating system interactions included and not simulations of those data streams;

b. Providing portability so that can be placed in a lap-top PC with appropriate interface cables;

c. Can be configured to work with multiple networking and communications technologies (RS-232, RS-422, Ethernet, etc.).

The simulations would be conducted at baseband and not include any effects of modulation. This is done for two reasons: it allows for simulating network channels other than space channels, and we are really interested in testing the performance of the networking protocols while the modulation provides an added layer of complexity to the simulation environment without providing more accurate results when looking at protocol performance. If there are modulation losses in the system, the bit error rate and statistical descriptions can be adjusted to match the expected performance without modulating the data explicitly.

SECTION III - VIRTUAL INSTRUMENT DESIGN

III.1 INTRODUCTION

In this section, we develop the design of the Virtual Instrument forming the heart of the SGLS channel error simulator. This will include a description of the error generation methodology as well as the programming to accomplish the error generation functions. A full description of the software modules to perform the necessary functions is also given.

As documented in [2], an initial hardware approach was developed to realize a methodology for generating the channel error profile. This initial development was based on a custom VME module that used a local disk file containing the error vectors. The module would perform the Exclusive-Or of the data with an error vector derived from a statistical generator developed in [3]. The error vector was selected by the user when the controlling C program was started and the vector was loaded from a disk file inside the VME chassis to the custom VME module. The data input was over one RS-422 connector on the VME module and the resulting modified data was output over another RS-422 connector on the VME module.

There were several problems with this approach. The major problem was that the VME module was uni-directional (forward or return link but not both without wire-wrapping another module). Therefore real protocol testing was not readily available. Secondly, the time-variable error generation based on a single simulated satellite pass did not work properly due to the C control program continuing to fault before completion. Additionally, this program was not well documented thereby making changes difficult. Finally, there was a hardware failure in the VME module. At this point, another approach was sought. The Virtual Instrument method appeared to be appropriate for the solution to the needs of the module development.

III.2 METHOD OF ERROR GENERATION

The error generation methodology used in the VI is the same as the one used in the hardware module. It is based upon the known relationship from digital logic that if one takes a digital data stream of logic 0s and 1s and then performs an Exclusive-Or (Ex-Or) operation on the data stream then every place where the data stream is Ex-Ored with a logic 0, the data is unchanged while every place where the data stream is Ex-Ored with a logic 1, the data symbol is complemented [4]. This can be used to model the channel error generation process: the channel can be modeled as an Ex-Or gate that randomly operates on the input data stream. This is illustrated in Figure 1 where a single bit error is generated in the output data stream.

Input Data Stream:	··· 0 0 0 1 1 0 0 1 ···
Error Vector:	··· 0 0 0 0 0 1 0 0 ···
Output Data Stream:	··· 0 0 0 1 1 1 0 1 ···
bit error location	⇑

Figure 1 - Channel error generation process.

To properly model a channel, the user needs a proper statistical description of the channel error generation mechanism. A typical channel error statistical description is Additive White Gaussian Noise (AWGN) where the errors are described by a Gaussian random process parameterized by the link Energy-per-bit-to-Noise-density ratio (Eb/No) [5]. Previous work at NMSU [3] generated a computer program whereby the user could specify an Eb/No value, the number of bit errors to be generated, and the type of statistics to be used and the program would produce a vector meeting this specification. The vector would be all 0s except for a 1 at the locations where the bit errors are to occur. The 1s would be distributed over the vector according to the statistics specified by the user. The program was designed to develop vectors for AWGN, radio frequency interference, and mixed noise-and-interference environments. Other statistical distributions could be generated by modifying the program to generate the desired statistical model. For all of the testing done here, the AWGN statistical model was used.

III.3 SELECTION OF VI METHODOLOGY

The Virtual Instrument was designed using the LabVIEW™ programing architecture. LabVIEW was chosen for the following reasons:

a. The programming language is available on PC, Macintosh, and UNIX platforms;

b. The programming language is object-oriented and allows for modular code development;

c. The programming language provides for convenient access to PC communications ports (RS-232 and Ethernet) for data flow through the modules.

LabVIEW is a graphical programming language that is data driven and not strictly sequence driven (it only operates on data as it becomes available). Additionally, LabVIEW manages all memory and I/O functions that normally the high-level language programmer would need to manage through programs and drivers.

The VI error generation module was designed to provide the following capabilities using the programming language primitives and built-in modules:

a. Allow for data flow in two directions simultaneously;

b. Allow user-selectable bit error rates for both data flow directions;

c. Allow bit error rate vectors to be pre-computed and loaded prior to data flow;

d. Use standard communications ports for data flow.

The general operation of the VI follows the following steps:

a. The user initializes the VI and sets ports for data input and output (baud rate and port number);

b. The VI reads each directional serial port to determine if data is present for processing;

c. The VI is to XOR the data with the error vector;

d. The VI writes the data modified by the errors to the appropriate directional data port;

e. The VI continuously loops as quickly as possible (no wait states: if no data available at the input port, loop back an poll again) to process the data with minim delay.

By investigating the capabilities of LabVIEW, it was evident that it would be able to support these operations.

III.4 VI COMPONENTS

The SGLS VI has two parts to it: the user interface and the programming language. In this section, we will describe the details of both components. Consulting the LabVIEW programming manuals may be necessary if the reader is not familiar with LabVIEW concepts.

III.4.1 User Interface

The user interface for the error generation VI provides the following features:

a. Select the communications port for the forward and return data links. For this module, the RS-232 communications port in the computer is used. The user decides if COM1 or COM2 is to be used for the forward or return link. LabVIEW designates COM1 as port 0, COM2 as port 1, etc. on the PC platform.

b. Select the baud rate for the forward and return links. Normally, standard RS-232 rates will be selected. Most PC communications ports support baud rates from 2400 bps through 115200 bps.

c. Provide the user with real-time indications of data flow. This is done by showing the input queue size on each communications port upon each program iteration.

d. Provide the user with a dialog box to select the desired bit error profile for the forward and return links. The current test configuration provides error files for Eb/No profiles in AWGN from 0.0 dB through 11 dB. The commonly-used files are listed in Table 1.

e. Provide the user with a run-time means to disable the software processing.

The user interface for the SGLS VI is illustrated in Figure 2. The input for the baud rate is done using the LabVIEW Text Tool on the panel. The forward and return data port can be selected by incrementing the selection slide using the Operating Tool. The software enable/disable is done using the toggle switch on the VI panel. This needs to set to the ON position prior to starting the VI operation. When the user has entered the data, set the enable switch to ON, then the LabVIEW execution is initiated by clicking the left-pointing arrow (⇨) on the command bar using the mouse.

Table 1. Typical Statistical Error Files for Use with AWGN			
I. 1000 bit errors per file			
File Name	**Target Eb/No (dB)**	**BER**	**File Size (K-Bytes)**
a825k.dat	8.25	0.0001315	929
a850k.dat	8.50	0.00007865	1553
a875k.dat	8.75	0.00005268	2318
a900k.dat	9.00	0.00002170	5626
a925k.dat	9.25	0.00001727	7069
a950k.dat	9.50	0.00001246	9798
a975k.dat	9.75	0.00000860	14193
a1000k.dat	10.00	0.00000477	25599
II. 100 bit errors per file			
File Name	**Target Eb/No (dB)**	**BER**	**File Size (K-Bytes)**
a825c.dat	8.25	0.0001271	97
a850c.dat	8.50	0.00008332	147
a875c.dat	8.75	0.00005388	227
a900c.dat	9.00	0.00002165	564
a925c.dat	9.25	0.00001741	702
a950c.dat	9.50	0.00001177	1037
a975c.dat	9.75	0.00000869	1405
a1000c.dat	10.00	0.00000474	2578
a1025c.dat	10.25	0.00000289	4216
a1050c.dat	10.50	0.00000298	4098
a1075c.dat	10.75	0.00000094	12925
a1100c.dat	11.00	0.00000095	12843

Table 1 (cont.). Typical Statistical Error Files for Use with AWGN			
III. 10 bit errors per file			
File Name	Target Eb/No (dB)	BER	File Size (K-Bytes)
a825d.dat	8.25	0.0001908	7
a850d.dat	8.50	0.0001605	8
a875d.dat	8.75	0.00004423	28
a900d.dat	9.00	0.00002935	42
a925d.dat	9.25	0.00001678	73
a950d.dat	9.50	0.00001237	99
a975d.dat	9.75	0.00000939	130
a1000d.dat	10.00	0.00000432	283
a1025d.dat	10.25	0.00000373	328
a1050d.dat	10.50	0.00000214	570
a1075d.dat	10.75	0.00000094	1304
a1100d.dat	11.00	0.00000082	1485
IV. Zero Errors Per File			
infinite.dat	∞	0	1

The program will then present the dialog box for the error file selection which is done using a standard Windows dialog box and can be selected with a mouse.

III.4.2 VI Programming

The SGLS LabVIEW program is divided into two sections: module initialization and the processing loop as illustrated in Figure 3. During the initialization phase, the user input is taken from the VI front panel and is passed to the serial port control elements. This includes setting the forward and return communications port numbers, and the communications baud rate. The serial port initialization assumes the following communications port parameters to be in place and changed by

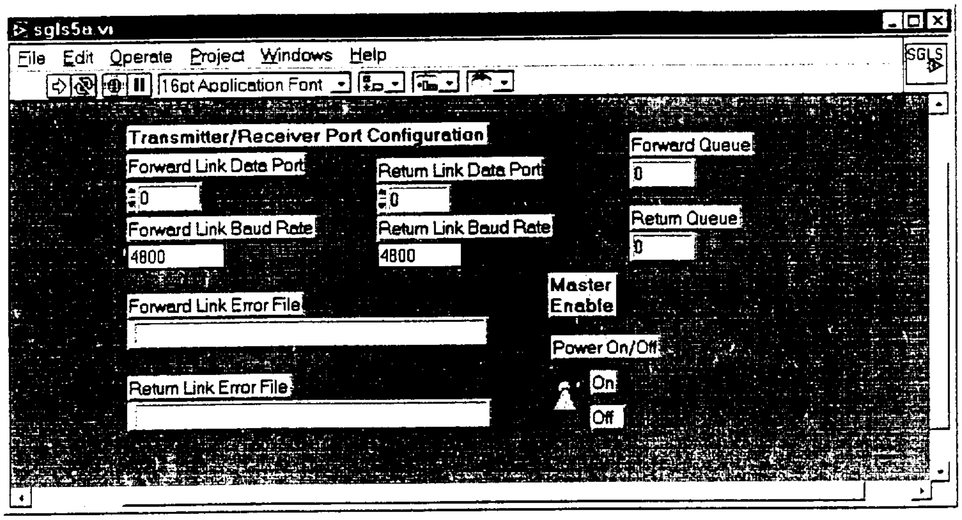

Figure 2 - User interface for channel error VI.

the user or the data sending device:

a. 8 data bits, 1 stop bit and no parity bits on each byte transferred,

b. No flow control is to be used to better simulate direct transmission through a radio channel, and

c. A null modem cable will be used to connect to the serial port (a straight-through cable will not work properly).

Because no flow control is used on the RS-232 port, a 16-K byte buffer is used to buffer the input data and keep from losing bytes. After setting the communications ports, the user is presented with a standard dialog box requesting the file specification for the forward and return link error vector files. The file path and name can be input directly or a mouse can be used to click through the selection of the drive, path, and file name.

The processing is controlled using a While Loop structure with no timing breaks and with continuous operation as long as the front panel toggle switch is in the ON position. The processing loop proceeds as follows:

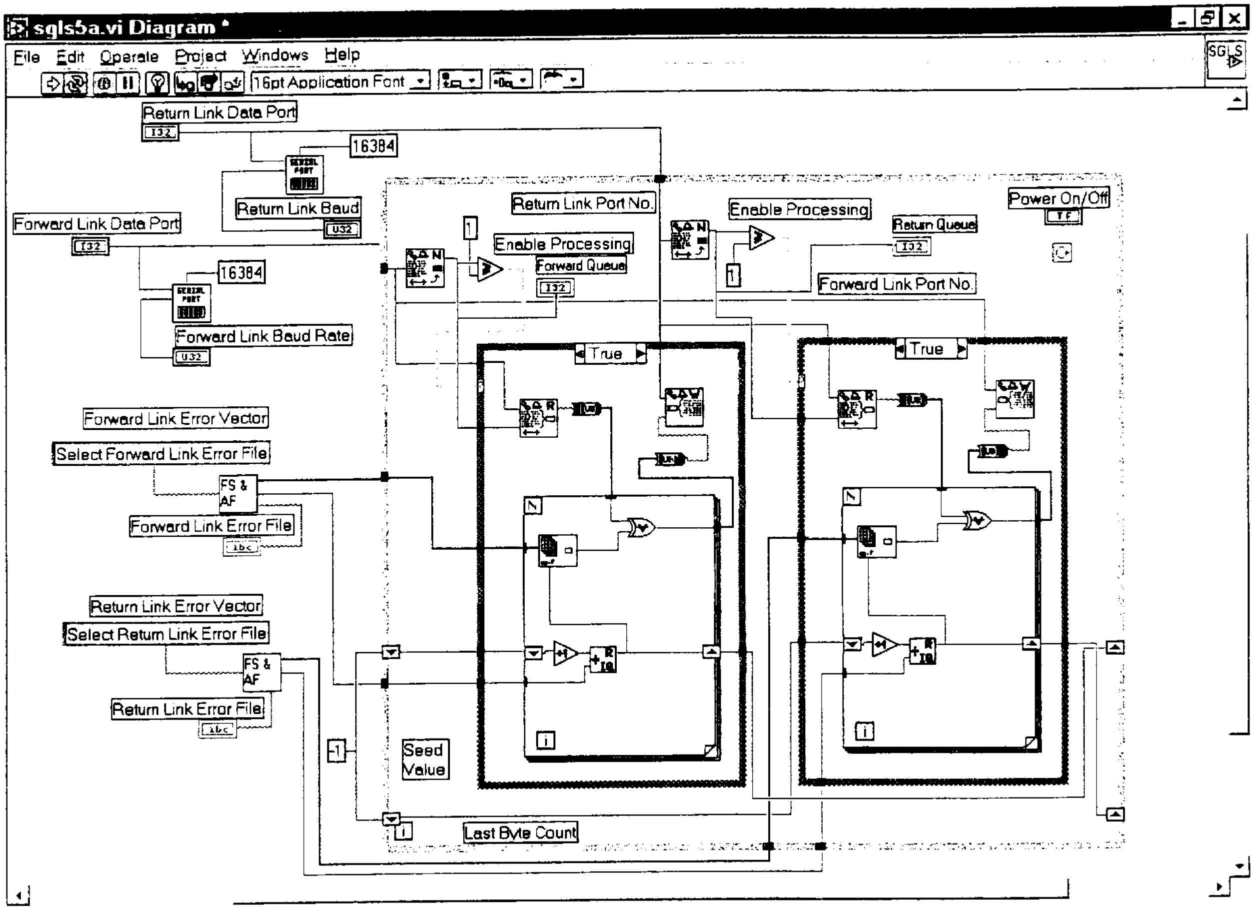

Figure 3 - Program for channel error VI.

a. Each input communications port is queried to determine if at least one byte of data is available (the loop only processes integer byte multiples of data) for processing,

b. For each port, if the port has no data to be processed, nothing is done for that port on the loop iteration.

c. If the port has data to be processed, all of the available bytes are read into the VI and a variable type change is made from string type to unsigned integer type. This step does not perform any modification to the data but makes the data type compatible for further processing.

d. For each communications port having data on this iteration, the data are sequentially processed in a Do Loop over all of the input bytes that were read in. Each byte of data is Ex-

Ored with the next byte of the error vector and the position index along the error vector is incremented as each byte is processed.

e. As the index along the error vector is incremented, if it comes to the end of the error vector, then the index is reset to the start of the error vector.

f. After all of the input bytes have been Ex-Ored with the error vector, the variable type is changed back from unsigned integer to string type and written to the indicated output port.

g. The While Loop then starts the next iteration.

Processing will continue until the user either places the toggle switch on the VI front panel at the OFF position using the Operating Tool or when the user clicks on the LabVIEW stop button with the mouse.

SECTION IV - VIRTUAL INSTRUMENT VALIDATION

The basic SGLS instrument validation was performed by working with each component of the VI as a self-contained sub module and using the VI display interface options to place debug displays at each step of the way. With these debug options in place, the data flow was monitored for correct operation. Typical debug tests included

a. Validation of the stepping through the error vector indices and proper roll over to the start of the vector when the end-of-vector count is reached;

b. Monitoring the input queue size to verify that it did not exceed 16384 bytes at which point data can be lost;

c. Verification of the Exclusive-Or operation by sending individual characters through the VI and monitoring the corrupted character results.

A typical throughput test of the VI compared the effective transfer rate to send a 76 KB file using a PC Hyperterminal data transfer test. In this test, the XMODEM protocol was used to transport the file through both the channel error simulator with the channel error rate set to zero errors (the processing continued but the error vector was all 0s) and via a direct null modem connection. The results of this test are shown in Table 2. Generally, the VI made the process run a bit slower but the queue was always bounded in length. From this we conclude that the VI presented no significant degradation to the transfer process.

Table 2. VI XMODEM Throughput Test

Baud Rate	Straight Through	VI in the Loop	VI Max. Queue Size
9600	7880 bps	7150 bps	< 10 B
19200	15100 bps	13200 bps	< 10 B
38400	23200 bps	23100 bps	< 10 B
57600	29400 bps	30500 bps	< 10 B
115200	31100 bps	30700 bps	< 100 B

A second timing validation test was run using the actual computers and protocols that would be used in the protocol testing. In this test, various files were sent using the TCP/IP ftp service at different baud rates. The total time to transmit the files under the condition that the SGLS made no errors in transmitting the data (an error vector of all 0s is used so that the timing remains the same) is compared with the time to transmit the same files over a short, straight null modem cable. A plot of the results is shown in Figure 4. Here we can see that the curves for the file transfer times when using the SGLS and a null modem cable are virtually the same. There was a slight difference for 100 K-byte files but the differences in the mean times were less than the variations in the mean times. We conclude that the SGLS causes no significant transmission delay nor does it introduce any link errors of its own, e.g. dropping bytes.

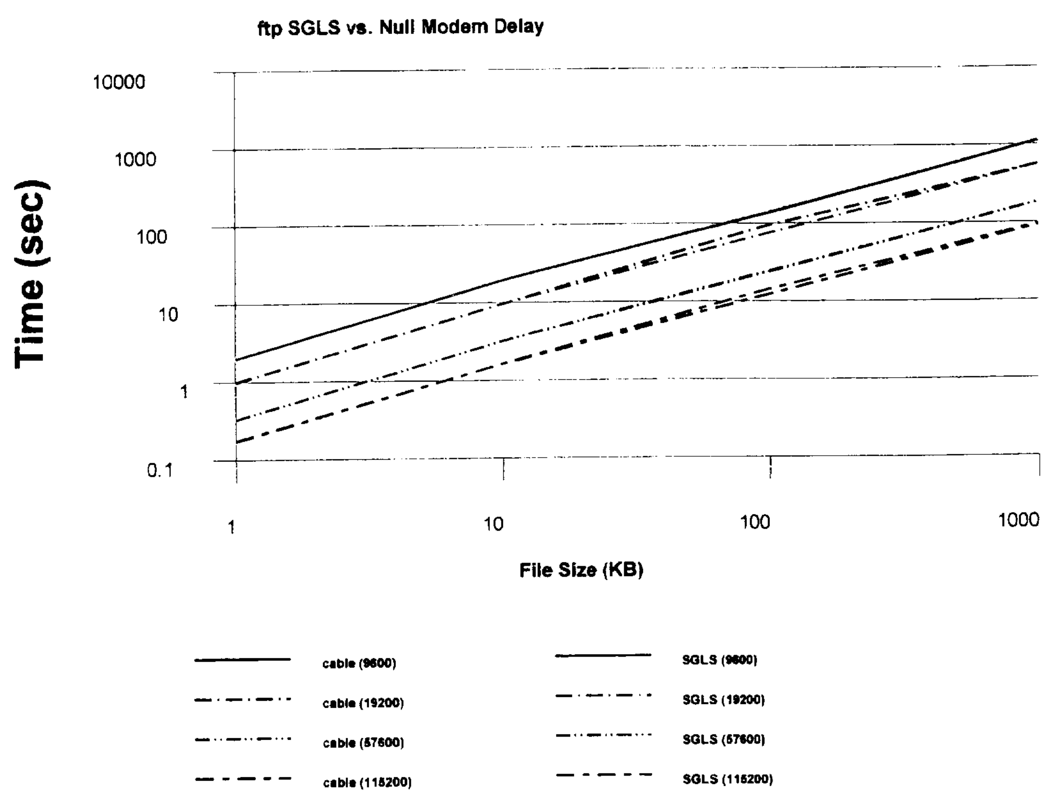

Figure 4 - Comparison of file transfer time between using the SGLS and a null modem cable for ftp file transfer services.

SECTION V - SAMPLE TESTS

V.1 TEST CONFIGURATION

The tests run with the SGLS were conducted using the configuration illustrated in Figure 5. The source and destination computers for the file transfer were two, identical Gateway PCs with 133 MHz processor speeds and 16 MB of memory running Red Hat Linux version 5.2. The computers were connected to the SGLS using commercially-obtained 6-foot null modem cables. Tests were run at channel Bit Error Rates (BER) of 0, 10^{-6}, 10^{-5}, and 10^{-4} using the files listed in Table 3. Files to be transferred were random text files having lengths of 1 KB, 10 KB, 100 KB and 1000 KB. For each file transmission test, ten runs were performed and the average time to complete the transmission recorded. In some of the tests at the high BER values, a transmission could not be completed due to the protocol timing out. These are noted in the file results. Measured data for all of the tests is given in the report Appendix.

Table 3. Error Vector Files Used in SGLS Transmission Tests

BER	Vector File
0	infinite.dat
10^{-6}	a1075d.dat
10^{-5}	a975d.dat
10^{-4}	a825c.dat

For each test run, the transmission rate in the forward and return direction was the same as was the BER on the forward and return rate.

V.2 FTP TESTS

The first battery of tests performed was the transmission of files using the TCP/IP ftp service with

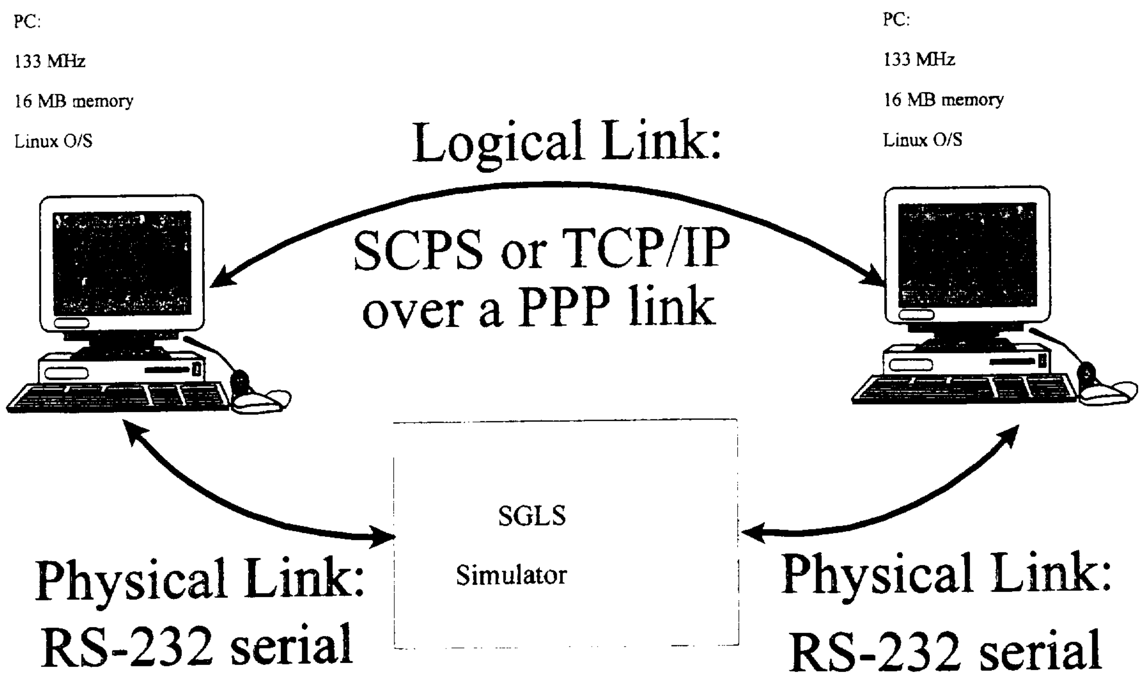

Figure 5 - Test configuration for TCP/IP and SCPS protocol testing.

the transmission error rates mentioned above. The results of these tests are summarized in Figure 6 where the transmission times for the various file sizes are displayed as a function of data rate and bit error rate. Each plot shows the transmission times for the 1 KB, 10 KB, 100 KB, and 1000 KB files with the 1-KB files taking the shortest time and the 1000-KB files taking the longest time. On each plot, the diamond marker on the y-axis represents the time to transmit the same file using the direct null-modem cable without the SGLS in the process. This is to give a reference indication of the best performance possible with these computers and operating system at the indicated data transmission rate. Interesting items noted during these tests include:

a. The file transfer process at a BER of 10^{-4} was generally not possible. In these cases, after many minutes of no activity on the link, the file transfer was aborted and restarted. The only file lengths that could be delivered were the 1-KB files. However, in each of the cases where delivery was possible, no test completed all ten experimental runs. The completion

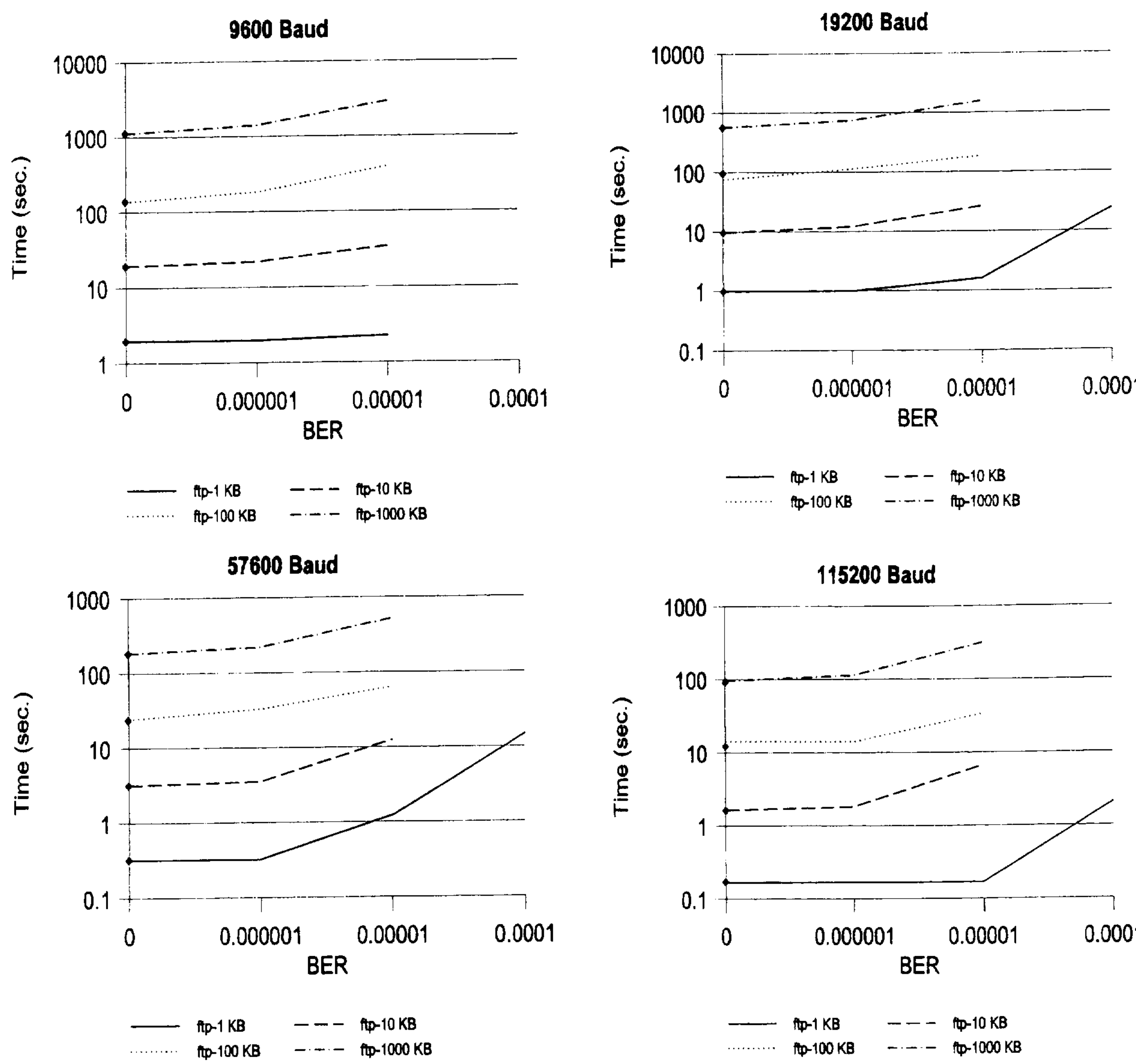

Figure 6 - File transmission time results using the ftp service as a function of BER and baud rate.

rates were

i. At 9600 baud, 0 of 10 experiment runs were completed,

ii. At 19200 baud, 8 of 10 experiment runs were completed,

iii. At 57600 baud, 2 of 10 experiment runs were completed, and

iv. At 115200 baud, 2 of 10 experiment runs were completed.

b. The file transfer process at a BER of 10^{-6} was nearly as good as the transfer process at a BER of 0. However, as the BER was increased to 10^{-5}, the transmission times rapidly increased as expected with TCP/IP confusing link errors for link congestion.

In all cases, TCP/IP was used as configured in the default Linux configuration and no attempt was made to vary parameters or otherwise tune the performance.

V.3 SCPS FP TESTS

The second group of tests performed was the transmission of files using the Consultative committee for Space Data Systems (CCSDS) Space Communications Protocol Specification (SCPS) File Protocol (fp) service [6] with the transmission error rates mentioned above. The SCPS-FP reference implementation we are using here is version 1.1.8 developed at MITRE [7] and is used with the default settings. The results of these tests are summarized in Figure 7 where the transmission times for the various file sizes are displayed as a function of data rate and bit error rate. As in the ftp results, each plot shows the transmission times for the 1 KB, 10 KB, 100 KB, and 1000 KB files with the 1-KB files taking the shortest time and the 1000-KB files taking the longest time. On each plot, the diamond marker on the y-axis represents the time to transmit the same file using the direct null-modem cable without the SGLS in the process. This is to give a reference indication of the best performance possible with these computers and operating system at the indicated data transmission rate.. Interesting items noted during these tests include:

a. The file transfer process at a BER of 10^{-4} was possible for the 1-KB. Again, for the longer files, the transmission was aborted after many minutes of no activity on the link. As in the TCP/IP experiments, in each of the cases where delivery was possible, no test completed all ten experimental runs. The completion rates were than TCP/IP and were as follows:
    i. At 9600 baud, 0 of 10 experiment runs were completed,
    ii. At 19200 baud, 8 of 10 experiment runs were completed,
    iii. At 57600 baud, 6 of 10 experiment runs were completed, and

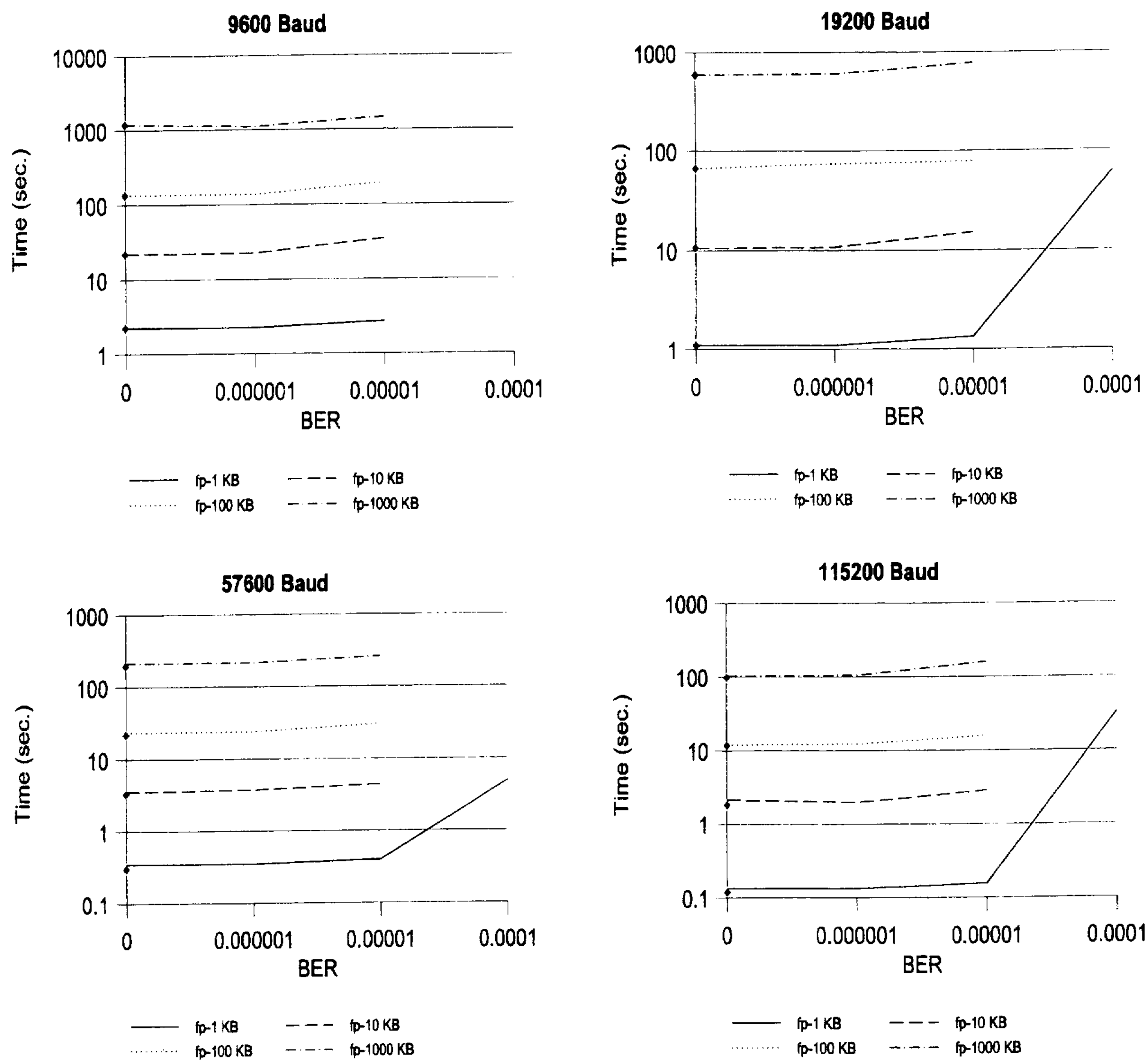

Figure 7 - File transmission time results using the fp service as a function of BER and baud rate.

iv. At 115200 baud, 5 of 10 experiment runs were completed.

b. As with the TCP/IP ftp service, the file transfer process at a BER of 10^{-6} was nearly as good as the transfer process at a BER of 0. However, as the BER was increased to 10^{-5}, the transmission times for SCPS fp did not show the same rapid increased as the TCP/IP ftp times did. This is expected due to the more appropriate way in which SCPS handles the channel errors and does not treat them as congestion and therefore slow down the link. Not all of the SCPS fp experiments were able to complete ten trials at a BER of 10^{-5}. This was a problem for the 100-KB and 1000-KB file lengths as follows:

i. At 9600 baud, only 9 of the 10 experiments with the 100-KB files completed,
ii. At 19200 baud, only 9 of 10 experiments completed with both the 100-KB and 1000-KB files, and
iii. At 115200 baud, only 9 of 10 experiments completed with the 1000-KB files.

In all experiments, the SCPS fp protocol parameters were left at the default settings provided by MITRE and no attempt was made to optimize the settings.

We show a comparison of the TCP/IP ftp service and the SCPS fp service transmission delay times in Figure 8. As we can see, at the low BER configurations, both ftp and fp have essentially the same transmission times. As the BER increases, the effects of the congestion algorithm in the TCP/IP ftp service can be seen because the transmission time rapidly increases at a BER of 10^{-5}. The SCPS fp protocol does a better job of maintaining a transmission time similar to the no-error case at this BER. The BER of 10^{-4} cases do not represent a good comparison because both protocols had great difficulty in maintaining a connection at this BER and the number of completed file transfers is very small.

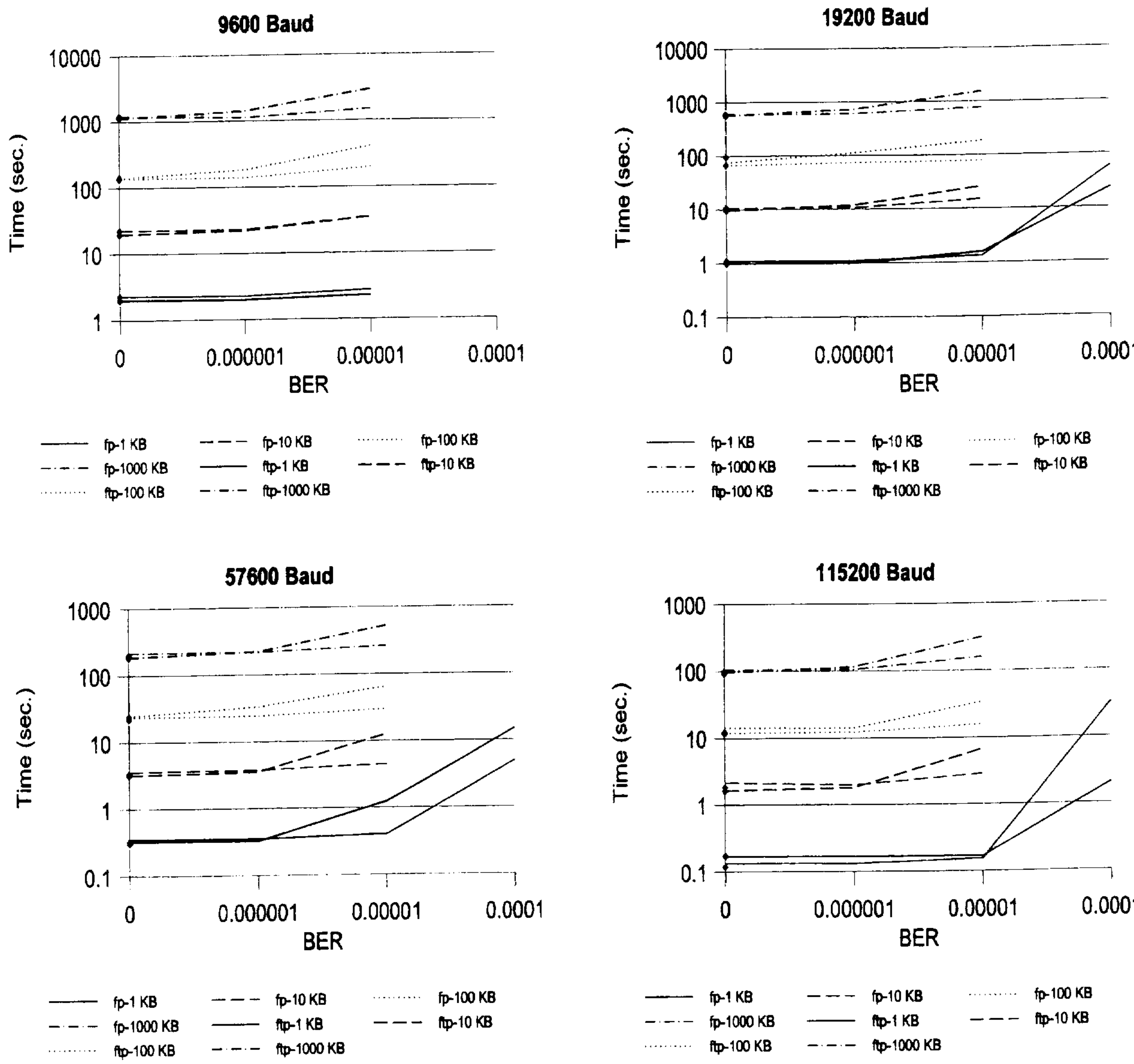

Figure 8 - Relative transmission times for ftp and fp as a function of file size and BER.

SECTION VI - SUMMARY AND CONCLUSIONS

A Virtual Instrument was constructed to realize a Space-to-Ground Link Simulator (SGLS) for performing baseband networking tests. In these initial tests the TCP/IP ftp and SCPS fp file transfer protocols were used with the SGLS simulator. Channel bit errors rates from 0 through 10^{-4} were used. The source and destination host computers were modest PC-class computers running the Linux operating system. The general results were found to be

a. Both protocols have transmission troubles at BER of 10^{-4}. The SCPS fp did better at file delivery in the large error environment in that a larger percentage of the 1-KB files were able to be transmitted but both protocols had problems in transferring files larger than 1 KB this error rate.

b. At low a BER of 10^{-6} or better, both protocols ran at about the same speed (to within statistical variations).

c. At a BER of 10^{-5}, the TCP/IP ftp protocol showed a significant degradation in performance in that a significantly longer transmission time was required than in the no-error case and longer than that required for the SCPS fp protocol. The SCPS protocol did show some trends not being able to complete a transmission at this BER with longer files than the TCP/IP ftp service did. However, with only 10 trials, this many not be a significant difference.

Based on these limited experiments, we conclude that both protocols work equally well in a low-error-rate environment. With bit error rates exceeding 10^{-6}, the SCPS fp protocol appears superior because the transmission time does not grow rapidly as does the TCP/IP ftp transmission time as the errors corrupt the packets. In high-error-rate environments, packets need to be kept short, approximately 1KB at most, to ensure a reasonable chance of data delivery.

SECTION VII - REFERENCES

[1] National Instruments™, LabVIEW™, version 5, Austin, Texas, January 1998.

[2] Lynde, W. H. and Horan, S., "Space Protocol Test and Evaluation Project," NMSU-ECE-96-004, April 1996.

[3] Moser, J. C. and Osborne, W. P., "Error Pattern Generation for Coded BPSK," NMSU-ECE-95-007, August 1995.

[4]Sklar, B., *Digital Communications Fundamentals and Applications*, Prentice-Hall: Englewood Cliffs, NJ, 1988, p.276.

[5] *ibid.*, p. 156.

[6] Consultative Committee for Space Data Systems, "Space Communications Protocol Specification (SCPS) - File Protocol (SCPS-FP)," CCSDS 717.0-R-3, September 1997.

[7] Feighery, P., private communication.

Appendix - Test Data

Protocol:	**FTP**		**Baud:**	9600	bps							
File Size:	1000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	1.955	0.00527	1.96	1.96	1.95	1.95	1.96	1.95	1.96	1.96	1.95	1.95
1E-06	1.96	0.008165	1.95	1.97	1.96	1.95	1.97	1.97	1.96	1.96	1.95	1.96
1E-05	2.245	0.897753	1.95	1.96	1.97	1.97	1.96	1.96	1.96	4.8	1.96	1.96
0.0001												

File Size:	10000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	19.1	3.6E-07	19.1	19.1	19.1	19.1	19.1	19.1	19.1	19.1	19.1	19.1
1E-06	21.57	5.207271	19.1	19.1	19.1	31.5	19.1	19.1	19.1	19.1	31.4	19.1
1E-05	34.29	9.091932	22.2	41.8	40.3	40.3	28.9	30.9	52.2	29	31.5	25.8
0.0001												

File Size:	100000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	136.2	1.932184	135	139	135	135	135	135	139	139	135	135
1E-06	179.6	59.70334	221	172	135	142	135	249	307	142	135	158
1E-05	395.4	135.5214	334	329	439	321	529	714	306	275	400	307
0.0001												

File Size:	1000000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	1109	30.71373	1090	1150	1090	1090	1090	1090	1090	1160	1090	1150
1E-06	1394	61.13737	1320	1260	1420	1430	1390	1400	1390	1460	1450	1420
1E-05	2895	77.63876	3030	2920	2970	2760	2850	2830	2940	2900	2840	2910
0.0001												

Protocol: FP **Baud:** 9600 bps

File Size:	1000	Bytes										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	2.209187	0.009452	2.209314	2.213607	2.207083	2.212135	2.223424	2.20383	2.224149	2.199009	2.195723	2.203591
1E-06	2.211429	0.008858	2.199526	2.212331	2.211627	2.200866	2.206857	2.226929	2.224419	2.213342	2.211258	2.207133
1E-05	2.664527	1.311604	2.199028	2.221749	6.383126	2.21163	2.211824	2.212473	2.216883	2.575144	2.211943	2.201474
0.0001												

File Size:	10000	Bytes										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	21.60859	0.056124	21.51071	21.58781	21.57742	21.73915	21.61442	21.61923	21.60949	21.61473	21.61438	21.59861
1E-06	22.21834	2.491341	21.63169	21.61407	21.64452	20.01195	29.16012	21.60441	21.63682	21.64292	21.61248	21.62442
1E-05	34.39882	15.58279	41.40141	28.4308	24.92662	32.32873	75.49184	28.34153	38.17912	24.9019	25.04917	24.93713
0.0001												

File Size:	100000	Bytes										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	131.5401	0.090209	131.5862	131.4791	131.5488	131.6789	131.7012	131.4513	131.5141	131.474	131.4463	131.5216
1E-06	133.919	1.862794	131.5167	135.1039	136.8612	131.6788	133.7312	131.6132	135.2237	135.0012	133.24	135.2204
1E-05	190.5651	61.7261	152.6101	188.0867	166.9746	155.777	287.6407	306.6148	155.32	149.0181	153.0437	
0.0001												

File Size:	1000000	Bytes										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	1171.02	4.247712	1170.734	1168.534	1169.115	1169.003	1169.918	1170.696	1169.846	1182.944	1169.487	1169.927
1E-06	1092.995	318.947	1195.483	1187.539	1186.841	186.5591	1186.538	1182.116	1239.407	1181.589	1199.386	1184.493
1E-05	1444.299	86.39444	1317.65	1391.403	1453.818	1567.237	1541.752	1321.493	1509.603	1478.491	1393.653	1467.888
0.0001												

Protocol:	FTP		**Baud:**	19200	bps							
File Size:	1000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	0.9759	0.005174	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.972	0.967	0.967
1E-06	0.979	1.3E-08	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979	0.979
1E-05	1.5647	1.23082	0.979	3.9	0.994	0.979	0.979	0.979	0.979	0.979	3.9	0.979
0.0001	23.931	32.08192	22	93.9	0.979	3.9	3.89	44.9	0.979	20.9		
File Size:	10000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	9.548	0.006325	9.54	9.54	9.55	9.54	9.55	9.56	9.55	9.55	9.55	9.55
1E-06	11.915	3.938912	9.55	9.55	9.55	16.2	9.55	9.55	16.2	9.55	19.9	9.55
1E-05	25.76	15.84173	14.7	16.2	29	55.2	16.1	14.2	53.7	19.9	14.7	23.9
0.0001												
File Size:	100000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	74.77	9.216417	71.8	71.8	71.9	71.8	71.9	71.8	101	71.9	71.9	71.9
1E-06	111.48	54.91255	207	74.9	99.1	71.8	123	71.9	64.3	208	71.8	123
1E-05	182.4	20.24955	166	164	178	179	172	187	236	181	185	176
0.0001												
File Size:	1000000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	562.9	16.01007	572	572	596	550	550	549	549	549	572	570
1E-06	728.3	85.65441	819	650	650	659	638	795	844	818	758	652
1E-05	1530	111.8034	1610	1580	1390	1710	1520	1640	1450	1470	1400	
0.0001												

Protocol: FP **Baud:** 19200 bps

File Size:	1000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	1.086338	0.027128	1.133237	1.062077	1.140565	1.072557	1.07361	1.075164	1.081078	1.07292	1.076097	1.076078
1E-06	1.075853	0.003859	1.076697	1.065829	1.078519	1.078567	1.075905	1.078222	1.074239	1.075196	1.078912	1.076448
1E-05	1.317525	0.706793	1.068604	1.061322	3.321895	1.073711	1.074564	1.069454	1.077187	1.247487	1.141255	1.039772
0.0001	61.18996	142.3591	3.312917	3.321405	1.499607	51.15239	5.33696	10.44628	3.321233	411.1289		

File Size:	10000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	10.3746	0.406548	10.13568	10.96371	10.1308	10.08482	10.13546	10.95531	10.16104	10.07651	10.96885	10.1338
1E-06	10.58653	0.837679	10.93735	10.06102	10.06172	12.65629	10.95869	10.06884	10.0616	10.06179	10.06078	10.93722
1E-05	14.92605	1.906756	16.92631	11.63053	14.70382	15.38809	16.9437	11.62803	14.70139	16.27043	15.37285	15.69533
0.0001												

File Size:	100000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	65.8194	0.41397	66.06931	65.08601	66.10184	66.07273	66.00565	65.98858	65.97155	64.99665	65.9592	65.94247
1E-06	72.87599	19.14698	66.92158	65.97606	67.64232	67.72933	66.80174	65.97624	66.83385	127.331	65.85621	67.69154
1E-05	77.12599	2.578465	75.54117	76.94615	74.99642	76.32667	83.59873	75.49623	76.64107	77.85322	76.73429	
0.0001												

File Size:	1000000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	584.5318	2.366544	586.7316	583.855	583.2174	584.5511	590.5022	583.248	583.2551	583.4883	583.2021	583.2676
1E-06	595.8796	4.486484	601.3274	592.4106	592.0493	593.0578	594.428	601.855	593.8116	603.5983	593.1989	593.059
1E-05	768.2053	74.27622	680.4842	844.9705	752.5099	701.6177	803.4728	674.9625	869.0469	746.0008	840.7823	
0.0001												

Protocol:	**FTP**		**Baud:**	57600	bps							
File Size:	1000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	0.32	0.002494	0.327	0.319	0.32	0.319	0.319	0.319	0.319	0.319	0.32	0.319
1E-06	0.3218	0.004237	0.332	0.32	0.32	0.327	0.32	0.32	0.32	0.32	0.32	0.319
1E-05	1.2214	1.434372	0.332	0.332	0.332	0.332	3.3	0.332	0.327	0.327	3.3	3.3
0.0001	14.925	8.449926	8.95	20.9								

File Size:	10000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	3.18	4E-08	3.18	3.18	3.18	3.18	3.18	3.18	3.18	3.18	3.18	3.18
1E-06	3.497	0.9884	3.19	6.31	3.18	3.19	3.19	3.19	3.18	3.18	3.18	3.18
1E-05	12.475	5.504613	5.77	12.9	13.2	13	9.99	6.49	13.5	12.3	26	11.6
0.0001												

File Size:	100000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	24.33	0.359166	23.9	24.2	24	24.3	23.9	24.8	24.8	24.3	24.3	24.8
1E-06	32.84	7.091811	30.1	40.8	41.2	32.6	24.3	25.8	29.6	41.4	24.2	38.4
1E-05	63.69	10.79583	60.2	75.8	56	57.4	58.6	56.8	89.4	60.7	65.1	56.9
0.0001												

File Size:	1000000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	183.2	0.632456	183	183	183	183	183	183	185	183	183	183
1E-06	218.9	7.880355	207	217	218	218	216	218	214	221	238	222
1E-05	521.3	41.37108	472	485	521	508	479	607	501	546	546	548
0.0001												

Protocol: FP Baud: 57600 bps

File Size:	1000	Bytes										
Error Rate	Avg	SDEV					Data	Time (sec)				
0	0.347105	0.008356	0.327429	0.347382	0.350061	0.338786	0.357654	0.350049	0.350083	0.347472	0.352276	0.349859
1E-06	0.344012	0.011743	0.338718	0.370065	0.346605	0.33748	0.335415	0.334286	0.346556	0.337492	0.335513	0.357986
1E-05	0.39342	0.155687	0.320649	0.33846	0.830978	0.340896	0.411812	0.33897	0.331981	0.343825	0.337185	0.339448
0.0001	4.915727	8.804056	0.779639	3.67153	0.812013	22.72204	1.182364	0.326775				

File Size:	10000	Bytes										
Error Rate	Avg	SDEV					Data	Time (sec)				
0	3.514524	0.017732	3.505196	3.49987	3.517891	3.505242	3.478406	3.517435	3.53568	3.531006	3.533345	3.521164
1E-06	3.708513	0.515661	3.501769	3.47979	3.479608	3.529125	5.098627	3.497487	3.493521	3.497673	3.487488	4.020038
1E-05	4.397189	0.411899	4.064699	4.479625	4.544508	4.049383	4.947471	4.017762	4.634713	5.124497	4.049792	4.05944
0.0001												

File Size:	100000	Bytes										
Error Rate	Avg	SDEV					Data	Time (sec)				
0	23.1524	0.024756	23.14606	23.21508	23.14989	23.15149	23.1635	23.15946	23.13027	23.12905	23.13855	23.1406
1E-06	23.65576	0.62797	23.85423	23.20963	23.45629	23.17604	23.32538	23.60884	23.50712	25.33112	23.34989	23.73905
1E-05	29.69662	5.008396	36.99914	26.86594	27.23349	32.32969	27.55222	39.98096	26.57158	26.41572	26.19634	26.82114
0.0001												

File Size:	1000000	Bytes										
Error Rate	Avg	SDEV					Data	Time (sec)				
0	210.5221	0.355727	210.3185	210.9331	210.9374	210.6861	210.8763	210.7	210.104	210.5128	210.0813	210.0718
1E-06	213.4107	0.625557	212.9783	213.0246	214.7074	212.7073	213.2223	213.5687	212.9287	213.6697	213.1328	214.1674
1E-05	258.229	20.06435	246.4816	280.2263	269.3509	236.7853	261.4706	239.4611	297.999	236.9375	253.9675	259.6098
0.0001												

Protocol:	**FTP**		**Baud:**	115200	bps							
File Size:	1000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	0.1693	0.020726	0.227	0.167	0.16	0.16	0.16	0.172	0.16	0.16	0.16	0.167
1E-06	0.1682	0.010799	0.179	0.172	0.172	0.192	0.167	0.16	0.16	0.16	0.16	0.16
1E-05	0.1685	0.002415	0.167	0.167	0.167	0.172	0.167	0.167	0.167	0.172	0.167	0.172
0.0001	2.086	1.575434	3.2	0.972								
File Size:	10000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	1.624	0.028363	1.7	1.62	1.61	1.61	1.61	1.61	1.61	1.61	1.64	1.62
1E-06	1.808	0.612097	3.55	1.62	1.61	1.61	1.62	1.61	1.61	1.61	1.62	1.62
1E-05	6.696	3.172143	5.55	5.52	3.55	4.86	2.77	7.41	8.38	14.1	7.41	7.41
0.0001												
File Size:	100000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	14.21	2.128876	12.4	16.2	16.2	11.7	12.1	12.4	16.3	16.2	16.2	12.4
1E-06	14.08	2.062792	11.9	12.4	12.6	15.7	16.2	12	16.1	16.2	15.9	11.8
1E-05	33.98	4.115769	31.9	28	32	41.7	32	32.9	37.7	38.9	31.9	32.8
0.0001												
File Size:	1000000	**Bytes**										
Error Rate	**Avg**	**SDEV**					**Data**	**Time (sec)**				
0	95.29	2.568376	93	93	93	93.1	98.3	98.1	93.1	98.2	98.2	94.9
1E-06	114.2	6.89283	108	114	112	116	114	132	111	113	107	115
1E-05	312.2	13.20606	298	319	322	325	302	325	301	308	329	293
0.0001												

Protocol: FP **Baud:** 115200 bps

File Size: 1000 **Bytes**

Error Rate	Avg	SDEV					Data	Time (sec)				
0	0.133643	0.007957	0.11279	0.139817	0.134914	0.135446	0.142687	0.135253	0.135388	0.13251	0.132556	0.135068
1E-06	0.132164	0.005182	0.117842	0.133686	0.1324	0.132372	0.13542	0.135285	0.132741	0.134969	0.132402	0.134522
1E-05	0.153617	0.092693	0.143615	0.143128	0.143676	0.385873	0.151231	0.131259	0.004241	0.143805	0.145625	0.143718
0.0001	32.95722	60.61326	24.06358	139.8109	0.133612	0.776173	0.001853					

File Size: 10000 **Bytes**

Error Rate	Avg	SDEV					Data	Time (sec)				
0	2.149799	0.320144	1.955256	1.942537	1.95499	2.612743	1.952436	2.614983	1.94529	1.954039	2.613356	1.952364
1E-06	1.970344	0.090019	1.942378	1.944738	1.939353	1.930759	1.935303	2.224114	1.931654	1.947377	1.973543	1.934224
1E-05	2.865828	0.87752	2.723725	2.21839	3.987491	4.148035	2.462904	4.183749	2.22867	2.203544	2.040338	2.461435
0.0001												

File Size: 100000 **Bytes**

Error Rate	Avg	SDEV					Data	Time (sec)				
0	11.85176	0.012792	11.83901	11.84416	11.84482	11.83496	11.83937	11.86739	11.85473	11.86189	11.86937	11.86189
1E-06	12.16152	0.444609	12.70967	11.84994	11.60822	12.08078	11.84507	12.30949	12.08326	11.6572	12.85168	12.61986
1E-05	15.72173	3.936166	13.94259	16.53157	26.19897	13.69604	13.22759	13.20558	15.35958	15.05213	13.05547	16.94782
0.0001												

File Size: 1000000 **Bytes**

Error Rate	Avg	SDEV					Data	Time (sec)				
0	102.4065	0.265355	102.6343	102.6744	101.7535	102.4322	102.2847	102.4579	102.6114	102.2808	102.4781	102.4579
1E-06	104.1989	0.65248	104.3762	105.7727	104.1622	103.7205	104.4248	104.5343	103.4741	103.7265	103.8787	103.9187
1E-05	157.6377	31.64371	167.7461	147.0604	162.6331	129.8817	161.4882	126.291	154.5618	232.2197	136.8578	
0.0001												